УДК 681.518

Д.Е. Иванов[1], *Р. Зуауи* [2]
[1]Институт прикладной математики и механики НАН Украины, г. Донецк, Украина
[2]Донецкий национальный технический университет, г. Донецк, Украина
ivanov@iamm.ac.donetsk.ua, zouaoui_ram@hotmail.com

Алгоритм построения инициализирующих последовательностей цифровых схем, основанный на стратегии симуляции отжига

В статье предлагается новый алгоритм построения инициализирующих последовательностей цифровых схем, который основан на стратегии симуляции отжига. Алгоритм использует итерационное улучшение свойств одного потенциального решения, называемого конфигурацией. В качестве конфигурации используются дискретные входные последовательности. Функция оценки строится на основе моделирования работы заданной схемы на оцениваемой последовательности. Эффективность алгоритма показана путём апробации на контрольных схемах ISCAS-89.

Введение

В разработке цифровых схем фаза тестирования является наиболее дорогостоящей. Она состоит из нескольких последовательных фаз, первой из которых является инициализация схемы. Она заключается в переводе цифровой последовательностной схемы из полностью неопределённого состояния в такое, когда элементы состояний принимают известное значение.

Существует два подхода к решению данной задачи: аппаратный и программный. В первом случае в схеме предусматривается аппаратный сброс, для чего добавляется избыточная аппаратура, которая реализует его механизм. По сигналу «сброс» данная аппаратура записывает в элементы состояний необходимые значения, соответствующие спецификации начального состояния.

Однако данный подход не применим к уже разработанному дизайну и поэтому часто используется программная инициализация. В этом случае задача заключается в построении входной последовательности, которая переводит цифровую схему в начальное рабочее состояние.

При решении задач построения идентифицирующих входных последовательностей выделяются три основных направления. К первому относятся структурные методы [1], которые основаны на построении деревьев решений и обходов по ним. Второе направление представлено методами, которые работают с символьным представлением булевых функций, реализующих схему [2], [3]. Оба данных направления показывают хорошие результаты со схемами малой и средней размерности. Однако для больших схем они плохо применимы ввиду того, что вызывают переполнения стеков в процедурах поиска обходов. В качестве альтернативы этим подходам образовался третий, который основан на том, что необходимые для поиска решения данные получаются на основе моделирования работы цифровой схемы [4], [5]. Это позволяет алгоритмам такого типа получать решения для схем большой размерности, поскольку задача моделирования является более простой и решена для больших схем.

Также следует отметить, что ряд методов построения инициализирующих последовательностей были внедрены непосредственно в алгоритмы генерации тестов и отдельно не описывались.

Среди глобальных оптимизационных стратегий можно выделить алгоритм симуляции отжига (СО). Изучение данной стратегии только набирает силу. Оптимизационный подход в алгоритмах симуляции отжига является принципиально более простым, чем, например, в генетических алгоритмах (ГА) [6]. Это объясняется тем, что в ней улучшение свойств происходит только для одного потенциального решения. Следовательно, исчезает громоздкий механизм обработки и построения популяций. Это позволяет, с одной стороны, эффективно применять данную стратегию к решению практических задач, а с другой стороны, построение решающих алгоритмов на её основе будет более простым.

В данной статье авторы предлагают применить стратегию СО для построения инициализирующих последовательностей цифровых последовательностных схем, а также исследовать её поисковые способности в сравнении с соответствующим генетическим алгоритмом, который был предложен авторами ранее [5].

Данная статья имеет следующую структуру. В первом разделе мы кратко опишем оптимизационную стратегию симуляции отжига. Во втором разделе мы сделаем формальную постановку задачи логической инициализации и опишем алгоритм СО её решения. В третьем разделе мы опишем эвристические методы, направленные на повышение эффективности предложенного алгоритма, и приведём результаты апробации. В четвертом разделе с целью сравнения поисковых свойств применительно к нашей задаче проводится сравнение работы генетических алгоритмов и стратегии симуляции отжига. В заключении будут сделаны выводы и предложены направления дальнейших исследований.

1. Оптимизационная стратегия симуляции отжига

Стратегия СО была предложена на заре численных экспериментов в [7] и относится к области слабого искусственного интеллекта. Авторы использовали подход для нахождения наиболее вероятного состояния группы атомов при остывании слитка металла. Предполагается, что процесс протекает при понижающейся температуре, атомы уже выстроились в кристаллическую решётку, однако их переход между состояниями ещё возможен. Вероятность таких переходов зависит от температуры, а устойчивое состояние соответствует минимуму энергии системы. Именно из этой работы заимствованы термины данной стратегии, связанные с металлургией: симуляция отжига, конфигурация, энергия, расписание температуры. Только по прошествии 30 лет в [8] было показано, что такой подход может быть использован как оптимизационная стратегия в общем виде.

Перед описанием алгоритма СО дадим определение некоторых терминов, которые в нём используются.

Под конфигурацией K_i шага алгоритма i понимается некоторая единичная точка в пространстве решений. Для конфигурации K_i возможно строить окружения – некоторое множество точек, которые получаются из K_i путём применения возмущающих операций. Для каждой конфигурации K_i (точки в пространстве решений) в соответствие ставится функция стоимости $C_i = C(K_i)$, которая показывает её качество. Распределение температур – последовательность фиксированных понижающихся температур $\{T_l\}$.

Тогда алгоритм симуляции отжига заключается в нахождении оптимума функции $C(K)$ и может быть описан следующим образом.

1. Алгоритм начинает работу с формирования начальной конфигурации K_0, после чего происходит вычисление её функции оценки: $C_0 = C(K_0)$. Начальная конфигурация принимается в качестве текущей конфигурации $K_i = K_0$. Далее выбирается первый элемент последовательности температур $\{T_i\}$ и определяется начальная температура $T_i = T_0$. Далее итеративно вплоть до выполнения условия остановки выполняются следующие шаги.

2. С помощью операции возмущения строится окружение текущей конфигурации K_i. Число конфигураций, которые входят в окружение, задаётся предварительно. После построения окружения, для всех конфигураций, в неё входящих, выполняются шаги 3 – 6.

3. Вычисляется изменение функции стоимости для некоторой конфигурации $K_{пром}$ из окружения:

$$\Delta C_i = C(K_{пром}) - C(K_i)\,.$$

4. Произведённые возмущением изменения либо принимаются, либо отвергаются: если изменение функции стоимости отрицательное, то промежуточная конфигурация заменяет текущую: $K_{i+1} = K_{пром}$. В противном случае такая замена происходит на основании распределения Больцмана:

$$K_{i+1} = \begin{cases} K_{пром}\,, если\, \Delta C_i < 0; \\ K_{пром}\;\; с\, вероятностью\, P = \exp(-\Delta C_i / kT_i), если\, \Delta C_i > 0; \end{cases}$$

где k – эвристическая константа.

Такой способ приёма изменений конфигурации позволяет при большей температуре чаще принимать плохие решения, чем отбрасывать их. При снижении температуры также снижается вероятность принятия худших решений.

5. Изменяется текущая температура:

$$T_{i+1} = обновить(T_i)\,.$$

6. Пока не достигнута конечная температура или предельное число итераций, переход к шагу 2.

Алгоритм завершает работу в двух случаях:

– на какой-либо итерации найдено точное решение задачи;

– достигнуто предельное число итераций, в этом случае решением является последняя исследованная конфигурация.

Применение алгоритма симуляции отжига для решения некоторой прикладной проблемы означает необходимость определения его компонент исходя из предметной области задачи.

2. Алгоритм построения инициализирующих последовательностей

При тестировании схем на вентильном уровне используется модель синхронных последовательностных схем (рис. 1).

В данной модели схема представляется в виде комбинационного блока и блока памяти, который состоит из D-триггеров. Далее также будем использовать следующие обозначения: $\#Вх$ – число внешних входов схемы, размерность вектора v; $\#Вых$ – число внешних выходов схемы, размерность вектора Y; $\#Тр$ – число элементов состояния (D-триггеров) схемы, размерность вектора T.

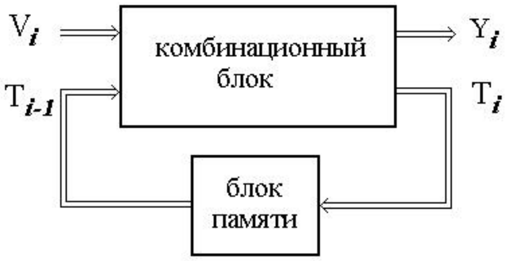


Рисунок 1 – Модель синхронной последовательностной схемы

Вектор v – упорядоченное множество двоичных значений, которое подаётся на вход цифровой схемы в определённый такт времени. Последовательность s_i заданной длины l_i – упорядоченное множество из l_i векторов, которые подаются на вход схемы в последовательные такты времени. Обозначение v_{ij} говорит, что мы рассматриваем в последовательности s_i вектор с номером j ($j = 0,...,l_i - 1$).

При моделировании используется преобразование синхронной последовательностной схемы в псевдокомбинационный эквивалент с дальнейшим его итеративным моделированием в последовательные такты времени. Для такого преобразования удаляют элементы состояний. Входы элементов состояний (вектор T_i на рис. 1) при этом называются псевдовходами, а их выходы (вектор T_{i-1} на рис. 1) – псевдовыходами.

При таком преобразовании работа блока памяти представляется следующим образом. В момент смены тактового импульса в устройстве синхронизации (для упрощения на рисунке не представлен) происходит подача на вход схемы новых входных значений v_i момента времени i, на псевдовходы подаются значения псевдовыходов схемы в предыдущий такт времени T_{i-1}, после чего путём моделирования комбинационной части схемы для такта времени i формируются выходные сигналы схемы Y_i и сигналы псевдовыходов T_i.

В данной работе мы будем рассматривать задачу логической инициализируемости схемы. Задача состоит в нахождении одиночной последовательности s, которая переводит схему из неопределённого состояния в некоторое известное состояние.

Дадим более формальное определение. Пусть Q – множество всех состояний последовательностной схемы, тогда $Q = \{0,1,X\}^{\#Tp}$ ($0,1,X$ – элементы трёхзначного алфавита моделирования работы схемы). Пусть S – множество всех возможных определённых состояний схемы, состоящей из $\#Tp$ триггеров: $S = \{0,1\}^{\#Tp}$. Пусть также Σ – множество всех возможных входных последовательностей s_i. Очевидно также, что для выбранного трёхзначного моделирования $S \subset \Sigma$. Пусть $x \in Q$ – начальное неопределённое состояние. Функция $F: Q \times \Sigma \to Q$ обозначает все состояния, достижимые схемой при поступлении на её вход последовательностей из Σ при использовании трёхзначного алфавита моделирования. Тогда некоторая последовательность $s \in \Sigma$ называется инициализирующей для заданной схемы, если в финальном состоянии $q = F(x,s)$ все состояния определены, т.е. $q \in S$.

Для адаптации алгоритма СО к решению поставленной задачи необходимо определить его компоненты: конфигурацию и окружение, функцию оценки, возмущающие

операции. Сходность стратегии с генетическими алгоритмами [5] позволяет заимствовать ряд определений, которые фактически являются стандартными в недетерминированных алгоритмах построения входных последовательностей.

В качестве конфигурации выбирается отдельная входная последовательность s_i, длина которой заранее не определена, а ширина каждого входного набора соответствует числу внешних входов схемы $\# Bx$.

В качестве функции оценки входной последовательности s_i выбирается взвешенное число следующих параметров.

1. Отношение числа инициализированных триггеров к их общему числу n_1, данный параметр напрямую показывает качество последовательности: чем выше данное отношение и ближе к единице, тем выше качество заданной последовательности s_i.

2. В случае, когда у двух последовательностей параметр n_1 одинаков, следует учитывать также активность вентилей при моделировании: чем выше активность, тем выше вероятность распространить ненулевые значения с внешних входов на элементы состояний. Параметр n_2 показывает активность вентилей в схеме при моделировании на заданной входной последовательности s_i.

3. Параметр n_3 – длина последовательности s_i, он необходим для того, чтобы при прочих равных условиях выбирать более короткие последовательности.

Таким образом, оценочная функция имеет следующий вид:

$$f(s) = f(n_1, n_2 n_3) = (c_1 * n_1 + c_2 * n_2) * c_3^{n3}, \qquad (1)$$

где смысл параметров n_1, n_2, n_3 описан выше, а константы c_1, c_2, c_3 подбираются экспериментальным путём.

Параметры n_1, n_2 в (1) получаются на основе моделирования работы исправной цифровой схемы на заданной последовательности s_i. Таким образом, данный алгоритм относится к группе алгоритмов поиска, которые основываются на моделировании. Это позволяет ему работать со всеми схемами, для которых инструментальная ЭВМ позволяет выполнять исправное моделирование.

Возмущающие операции для выбранного кодирования конфигураций также заимствуются из ГА построения входных последовательностей [9]: добавление входного вектора в случайную позицию, удаление входного вектора из случайной позиции, случайное изменение входного вектора. Смысловая нагрузка данных операций остаётся такой же, и для краткости мы не будем останавливаться на данном моменте.

С определёнными таким образом компонентами мы реализовали предложенный алгоритм программно на языке программирования C++. Для апробации программной реализации были проведены машинные эксперименты на последовательностных схемах из международного каталога ISCAS-89 [10]. Для моделирования работы исправных схем и получения информации, необходимой для вычисления функции оценки, использовался алгоритм событийного моделирования [11]. Объём программной реализации вместе с алгоритмами моделирования составил около 1500 строк кода.

Эвристические параметры экспериментов составили: начальная температура T_0 = 120, конечная температура $T_{кон}$ = 1, распределение Больцмана, максимальное число рассматриваемых конфигураций для одной температуры 100, число итераций без улучшения 50, константа Больцмана для средних схем k = 0,0001, для больших k = 0,000001. Результаты данных экспериментов приведены в табл. 1, колонка I. Поиск останавливался в трёх случаях:

– была найдена последовательность, которая инициализирует все элементы состояний;
– за данное число шагов не происходило улучшений генерируемых решений;
– достигнута конечная температура T_k .

Таблица 1 – Результаты машинных экспериментов для схем ISCAS-89

Имя схемы	Число триггеров в схеме	Число инициированных триггеров	Длина инициализирующей последовательности		Время работы генератора, с.		Рассмотрено вариантов при поиске	
			I	II	I	II	I	II
s344	15	15	2	2	0	7	4	11901
s349	15	15	2	2	0	7	2	11901
s382	21	21	1	-	0	-	3	-
s386	6	6	3	2	0	7	6	11901
s400	21	21	1	-	0	-	8	-
s444	21	21	1	-	0	-	4	-
s499	22	0	-	-	7	7	11901	11901
s510	22	0	-	-	7	7	11901	11901
s526	21	21	6	2	0	7	401	11901
s526n	21	21	6	2	0	7	401	11901
s635	32	12	1	-	8	-	11901	-
s641	19	19	1	-	0	-	2	-
s713	19	19	1	-	0	-	6	-
s820	5	5	1	-	0	-	2	-
s832	5	5	1	-	0	-	2	-
s838.1	32	0	-	-	7	7	11901	11901
s938	32	0	-	-	7	7	11901	11901
s953	29	10	1	-	11	-	11901	-
s967	29	10	1	-	11	-	11901	-
s991	19	0	-	-	7	7	11901	11901
s1196	18	18	1	-	0	-	5	-
s1238	18	18	1	-	0	-	2	-
s1423	74	74	4	3	0	9	12	11901
s1488	6	6	1	-	0	-	1	-
s1494	6	6	1	-	0	-	2	-
s1512	57	53	2	2	67	67	59501	59501
s3271	116	116	11	11	28	28	426	11901
s3330	132	132	5	3	0	14	698	11901
s3384	183	183	8	7	0	19	68	68
s4863	104	104	12	2	0	52	92	23801
s5378	179	179	110	14	40	55	7397	11901
s6669	239	239	7	7	2	28	1021	11901
s9234	228	52	1	-	61	-	23801	-
s13207	669	193	65	65	178	178	23801	23801
s15850	597	299	27	27	369	369	47601	47601
s35932	1728	1728	1	-	0	-	2	-
s38417	1636	372	55	55	292	292	11901	11901
s38584	1452	1383	58	58	2151	2151	35701	35701

Отметим также, что для данного набора схем не известны точные решения данной задачи. Однако разными авторами уже публиковались экспериментально полученные данные, в частности с помощью генетических алгоритмов [4], [5].

3. Оптимизация алгоритма

Алгоритм, описанный в предыдущем разделе, заканчивал свою работу сразу, когда была найдена любая последовательность, решающая задачу. Эксперименты показывают, что такие последовательности в большом количестве случаев имеют избыточную длину, и она может быть существенно уменьшена. Также в описанной версии вероятности операций возмущения оставались неизменными на протяжении всей работы алгоритма. В данном разделе мы опишем механизм оптимизации работы алгоритма, который направлен на поиск последовательностей с более короткой длиной и который для этой цели использует динамическое изменение значений вероятностей для операций возмущения.

Видно, что для описанного в разделе 1 алгоритма изменение вероятностей операций возмущения существенно изменяет направление поиска. Например, увеличение вероятности возмущения для удаления входного вектора направит алгоритм на поиск более коротких последовательностей за счёт увеличения времени поиска. Отсюда можно сделать вывод, что изменением значений вероятностей различных операций возмущения по фазам алгоритма можно добиться оптимального значения отношения «время поиска – длина последовательности» для конкретной задачи.

Нами предлагается модификация алгоритма, которая позволит улучшить его поисковые свойства за счёт динамической подстройки вероятностей операций возмущения непосредственно в процессе работы алгоритма.

Для оптимизации параметра распределения вероятностей между различными операциями возмущения был предложен следующий подход.

В процессе поиска инициализирующей последовательности условно выделены три фазы.

Фаза 1. Инициализация первого триггера. В начальный момент времени все триггеры находятся в неопределённом состоянии X. Очень важно для дальнейшей работы найти последовательность, которая «включит» схему, т.е. переведёт хотя бы один триггер в некоторое определённое состояние (0 или 1). Во время этой фазы важнейшей является операция возмущения, которая добавляет входные наборы в последовательность (т.е. увеличивает её длину). Данная операция должна применяться с наибольшей вероятностью. С другой стороны, в этой фазе нет необходимости в возмущении удаления входных векторов. Вероятность применения этой операции должна быть минимальной.

Фаза 2. Наращивание числа инициализированных триггеров. В этой фазе допустимо применение обеих указанных операций: и удаление входного вектора, и его добавление. Здесь происходит стандартный поиск оптимального решения.

Фаза 3. Уменьшение длины найденной последовательности. Обычно сгенерированные приближёнными методами входные последовательности имеют существенно избыточную длину: задача решается не всей последовательностью, а некоторой её подпоследовательностью. При этом также заранее не известен метод поиска такой подпоследовательности. В этой фазе необходимо отдать предпочтение операции удаления векторов. При этом алгоритм допускает с некоторой небольшой вероятностью, что качество последовательности может быть ухудшено. Поэтому данная фаза должна применяться уже при низкой температуре, что согласно применяемому распределению Больцмана существенно уменьшит вероятность принятия отрицательных решений. В этой фазе нет смысла использовать операцию добавления векторов, которая увеличивает длину входной последовательности. Данная третья фаза поиска применяется в двух случаях:

– инициализированы все элементы состояний;

– определённое число итераций не происходит улучшения качества входной последовательности. Поскольку данная ветвь алгоритма соответствует случаю, когда не все элементы состояний имеют определённое значение, то возможен переход от фазы 3 к фазе 2 (также происходит изменение распределения вероятностей операций мутации) в том случае, если произошло улучшение качества входной последовательности.

Для экспериментальной проверки высказанных предположений были предложены распределения вероятностей для трёх фаз алгоритма (табл. 2).

Таблица 2 – Вероятности операций возмущения для различных фаз алгоритма

	Распределение вероятностей			
	$P_{удаления}$	$P_{изм.столбца}$	$P_{изм.строки}$	$P_{добавления}$
Фаза 1	0.0	0.25	0.25	0.5
Фаза 2	0.25	0.25	0.25	0.25
Фаза 3	0.5	0.25	0.25	0.0

Соответствующие изменения были внесены в программную реализацию алгоритма. После этого проведена новая серия машинных экспериментов на том же наборе контрольных схем. В случае, когда для какой-либо схемы при использовании немодифицированной версии алгоритма найдена последовательность длины 1, вторая серия экспериментов для данной схемы не проводилась. Численные результаты, позволяющие произвести сравнение двух версий алгоритма, приведены в табл. 2 (колонка II).

В 6 случаях из 15 дополнительных экспериментов удалось уменьшить длину входной последовательности.

Ни в одном из 5 случаев, когда первый вариант алгоритма не нашёл инициализирующую последовательность, второй вариант за счёт увеличения глубины поиска также не смог произвести инициализацию хотя бы одного триггера. Эти данные полностью соответствуют опубликованным в [4] и, по-видимому, говорят о полной неинициализируемости данных схем.

4. Сравнение алгоритмов СО и ГА построения инициализирующих последовательностей

С целью сравнения поисковых свойств стратегии симуляции отжига с генетическими алгоритмами, было проведено сравнение с результатами работы, соответствующими алгоритму, описанному авторами в [5]. Аккуратное сравнение алгоритмов данного типа является сложной проблемой [12]. Определить абсолютно лучший алгоритм не представляется возможным, поскольку сравнение производится сразу по нескольким критериям: число инициализированных триггеров, время работы, длина полученной последовательности, число рассмотренных точек в пространстве поиска. При сравнении в качестве числовых данных для стратегии симуляции отжига в качестве числовых данных брались:

– данные для немодифицированной версии, если модификация не улучшала свойства найденной последовательности;

– данные для оптимизированной версии в противном случае.

Сводные данные результата сравнения алгоритмов построения инициализирующих последовательностей, основанных на ГА и СО, приведены в табл. 3. Мы не проводили сравнение по времени работы алгоритма в связи с тем, что инструментальные ЭВМ отличались при проведении серий экспериментов.

Таблица 3 – Сравнение поисковых свойств алгоритмов построения инициализирующих последовательностей, основанных на стратегиях ГА и СО

Параметр сравнения	Результаты сравнения		
	лучше ГА	лучше СО	ничья
число инициализированных триггеров	4	0	34
длина инициализирующей последовательности	1	3	34
рассмотрено число точек в пространстве поиска	14	18	6

По результатам сравнения можно сделать следующие выводы. Обе стратегии вполне успешно решают поставленную задачу построения инициализирующих последовательностей, причём показывают примерно одинаковые результаты по всем параметрам сравнения. Стратегия ГА несколько выигрывает по параметру «число инициализированных триггеров». Стратегия СО немного выигрывает по параметру «длина последовательности», причём в двух случаях из трёх – за счёт предложенной модификации. По параметру «рассмотрено число точек в пространстве поиска» выигрывает стратегия СО, причём в основном в случаях легко инициализируемых схем. Тогда как ГА за счёт внутреннего параллелизма лучше справляется с более сложными схемами.

Заключение

В статье описан алгоритм построения инициализирующих последовательностей для синхронных последовательностных схем. Он основан на новой оптимизационной стратегии симуляции отжига. В данном алгоритме происходит итеративное улучшение свойств одной предварительно полученной конфигурации. В качестве конфигурации выступает входная двоичная последовательность. Апробация программной реализации на схемах каталога ISCAS-89 показала высокую эффективность предложенного алгоритма. Также предложен эвристический метод распределения вероятностей операций возмущения, позволяющий в ряде случаев существенно уменьшить длину искомой инициализирующей последовательности. Проведено сравнение программной реализации алгоритмов СО и ГА построения инициализирующих последовательностей.

В качестве дальнейших исследований можно отметить применение стратегии СО для решения других задач построения идентифицирующих последовательностей: тестовых и верифицирующих эквивалентность заданных схем. Также интересной видится разработка параллельных (в том числе для многоядерных рабочих станций) версий алгоритмов идентификации, основанных на стратегии СО.

Литература

1. Niermann T. Patel J.H. HITEC: A Test Generation Package for Sequential Circuits / T. Niermann // Proc. European Design Automation Conf. – 1991. – P. 214-218.
2. Bryant R.E. Symbolic Boolean manipulation with ordered binary decision diagrams / R.E. Bryant // ACM Comput. Surv. – Vol. 24, № 3. – P. 293-318.
3. Sellers F.F. Analysing errors with the boolean difference / F.F. Sellers, M.Y. Hsiao, L.W. Bearnson // IEEE Transactions on Computers. – 1967. – № 5. – P. 675-680.
4. A New Approach for Initialization Sequences Computation for Synchronous Sequential Circuits / F. Corno, P. Prinetto, M. Rebaudengo [and others] // ICCD97, (October 1997, Austin, Texas (USA)). – P. 381-386.
5. Иванов Д.Е. Построение инициализирующих последовательностей синхронных цифровых схем с помощью генетических алгоритмов / Д.Е. Иванов, Ю.А. Скобцов, А.И. Эль-Хатиб // Проблеми інформаційних технологій. – 2007. – № 1. – С. 158-164.
6. Скобцов Ю.А. Основы эволюционных вычислений / Скобцов Ю.А. – Донецк : ДонНТУ, 2008. – 326 с.
7. Equation of State Calculation by Fast Computing Mashines / N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth [and others] // J. of Chem.Phys. – 1953. – Vol. 21, № 6. – P. 1087-1092.
8. Kirkpatrick S. Optimization by simulating annealing / S. Kirkpatrick, C.D. Gelatt, M.P. Vecchi // Science, 220. – 1983. – P. 671-680.
9. Иванов Д.Е. Генетические алгоритмы построения идентифицирующих последовательностей для цифровых схем с памятью / Д.Е. Иванов // Наукові праці Донецького національного технічного університету. Серія «Обчислювальна техніка та автоматизація». – Донецьк : ДонНТУ, 2008. – Вип. 14(129). – С. 97-106.

10. Brgles F. Combinational profiles of sequential benchmark circuits / F. Brgles, D. Bryan, K. Kozminski // International symposium of circuits and systems, ISCAS-89. – 1989. – P. 1929-1934.
11. Барашко А.С. Моделирование и тестирование дискретных устройств / Барашко А.С., Скобцов Ю.А., Сперанский Д.В. – Киев : Наукова думка, 1992. – 288 с.
12. Corno F. Comparing topological, symbolic and GA-based ATPGs: an experimental approach / F. Corno, P. Prinetto, M. Rebaudengo, M. Sonza Reorda // Proceedings of the IEEE International Test Conference on Test and Design Validity, (Washington (USA), October 1996). – P. 39-47.
13. Зуауи Р. Сравнение эволюционных поисковых стратегий: генетические алгоритмы и симуляция отжига // Материалы III Международной научно-практической конференции молодых учёных, аспирантов, студентов [«Сучасна інформаційна Україна: інформатика, економіка, філософія»], (Донецк, 14 – 15 мая 2009 г.). – С. 242-246.

Іванов Д.Є., Зуауї Р.

Алгоритм побудови ініціюючих послідовностей цифрових схем, що базується на стратегії симуляції відпалу

У статті пропонується новий алгоритм побудови ініціюючих послідовностей цифрових схем, який базується на стратегії симуляції відпалу. Цей алгоритм використовує ітераційне покращення якостей одного потенційного рішення, що називається конфігурацією. Функція оцінки будується на результатах моделювання без пошкоджень поведінки схеми на послідовності, що досліджується. Ефективність алгоритму перевірено шляхом апробації на контрольних схемах з каталогу ISCAS-89.

Ivanov D.E., Zouaoui R.

Algorithm for Initializing Test Sequences Generation of Digital Circuits, that is Based on the Simulating Annealing Strategy

In this paper a new algorithm for initializing test sequences generation is proposed. This algorithm is based on the new optimization strategy – simulated annealing. An iterative improvement of the one potential task’s solution is used. This potential solution is named configuration and is presented by the single input test sequence. The cost function is calculated on the basis of fault-free simulation that performs on the evaluated sequence. The effectiveness of the proposed algorithm is tested on the ISCAS-89 benchmark set.

Статья поступила в редакцию 03.07.2009.